B4

mapping mechanism is associated with the plurality of applications and is arranged to obtain the command object without directly involving the selected application.

REMARKS

Reconsideration of the rejections set forth in the Office Action dated July 21, 1999 is respectfully requested. Claims 1-20 have been rejected. New claims 21-23 have been added. Accordingly, claims 1-23 are currently pending.

Claims 1, 9, and 16-20 have been amended. Claim 1 has been amended to recite that a command object is obtained using substantially only the data associated with an application. Support for this amendment may be found in the Specification, as for example on page 8, at lines 8-11. Claim 9 has been amended to more clearly recite that an interface and a mapping mechanism are independent from an application. Support for this amendment may be found, for example, in the Specification on page 7, at lines 18-28. Claims 16-20 have been amended for clarity. Claim 16 has further been amended to recite limitations similar to those recited in claim 9, as amended.

New claim 21 requires that a command object is obtained without external input from a user of an application. Support for this new claim may be found in the Specification, as for example on page 8, at lines 8-11. New claim 22 recites that a command object may be obtained without directly involving an application. Support for this new claim may be found in the Specification, as for example on page 8, at lines 8-11.

New claim 23 requires that a framework for associating data with a command object which operates on the data which is associated with a selected application includes a data handler mechanism that is independent from and arranged to interface with a plurality of applications that includes the selected application. The framework also includes a data retriever mechanism that obtains the data and passes the data to the data handler mechanism, as well as a mapping mechanism that is arranged to obtain the command object, and is associated with the plurality of applications. Support for this new claim may be found, for instance, in the Specification from page 8, on line 15, to page 10, on line 13.

Rejections under 35 U.S.C. § 103

The Examiner has rejected claims 1, 2, 5, 6, 8, 9, 12-16, 19, and 20 under 35 U.S.C. § 103(a) as being unpatentable over "Navigating the Internet" by Gibbs in view of "Windows 95 Secrets" by Livingston. Claims 3, 4, 7, 10, 11, 17, and 18 have been rejected under U.S.C. § 103(a) as being unpatentable over "Navigating the Internet" by Gibbs in view of "Netscape Navigator" by Fulton.

Independent claim 9 recites a method for associating data with a command object in response to a request from an application. The method includes accessing data through an interface, accessing a mapping mechanism that is accessed by the interface and locates a command object, and obtaining the appropriate command object for the data. The interface binds the command object to the data, then returns the command object to the application. Both the interface and the mapping mechanism are independent from the application, *e.g.*, the interface and the mapping mechanism are not a part of the application. Though independent from the application, the interface is in communication with the application.

An interface which is separate from an application and accesses data, in addition to accessing a mapping mechanism to locate a suitable command object for the data, allows the application to be indirectly associated with command objects. Maintaining command objects independently from applications enables command objects, as well as data types, to be modified, created, and deleted without affecting the application (Specification, on page 7, at lines 15-18). Further, maintaining command objects independently enables one set of command objects to be used by multiple applications, thereby consolidating the number of command objects associated with a computer system and increasing the efficiency of a computer system.

It is respectfully submitted that none of the art of record teaches or reasonably suggests an interface and a mapping mechanism that are separate from the application. Specifically, Gibbs teaches accessing an attachment by saving the attachment into a file, and running uudecode (Gibbs, page 44). Saving an attachment into a file and then running a separate application to access the attachment does not teach or reasonably suggest accessing data , *e.g.*, an attachment, through an interface that communicates with an application and eventually returns a command object to the application. Hence, claim 9 is believed to be allowable for at least this reason.

In addition, claim 9 requires the binding of a command object to data using an interface. Contrary to the Examiner's assertions, Livingston does not appear to teach binding data to a command object. Specifically, Livingston teaches that one can choose the actions that will be taken on a file or document (Livingston, page 262). Choosing an action that will be taken on a file does not teach or suggest binding the action to the file, or binding a command object to data, as required by claim 9. Further, Livingston teaches of using a graphical user interface that is a part of an application like Windows 95 to choose an action to be taken on a file. Therefore, it is respectfully submitted that Livingston also does not teach the use of an interface that is independent from an application that requests a command object.

The Applicants submit that the registry as taught by Livingston provides a list of actions that an application can undertake (Livingston, page 262). However, the registry of Livingston does not actually undertake any actions, *e.g.,* the registry does not locate or obtain a command object. Further, the registry does not pass a command object to any interface, let alone an interface that is independent from an application. Therefore, it is respectfully submitted that a registry is not equivalent to, and does not reasonably suggest, the mapping mechanism of claim 9. Accordingly, claim 9 is believed to be allowable for at least this reason as well.

Claims 10-15 each depend either directly or indirectly from independent claim 9, and are, therefore, each believed to be allowable over the art of record for at least the reasons set forth with respect to claim 9. Each of these dependent claims recites additional limitations which, when considered in view of independent claim 9, further distinguish the claimed invention over the art of record.

Independent claim 16 recites a computer program product which performs the steps of claim 9, and is therefore believed to be allowable over the art of record for at least the reasons set forth above with respect to claim 9. As claims 17-20 each depend from claim 16, claims 17-20 are each also believed to be allowable for at least these reasons as well.

Claim 1 recites a framework for associating data, which is associated with an application, with a command object that is arranged to operate on the data. The framework includes a data handler mechanism that interfaces with the application, and a data retriever mechanism that communicates with the data handler to obtain data and to pass the data to the data handler mechanism. The framework also includes a mapping mechanism which obtains the command object based substantially on the data.

A framework that obtains a command object based on data, rather than based on a user selection or an application that will use the command object, effectively enables the application to be "blind" to the process of associating data with an appropriate command object (Specification, on page 7, at lines 16-23). Hence, command object may remain separate from applications. By maintaining command objects independently, data types and command types may be readily modified, created, and deleted (Specification, on page 7, at lines 25-28). Additionally, command objects may be used by multiple applications. As such, the efficiency with which command objects may be implemented and used is increased.

The Examiner has asserted that Gibbs in view of Livingston teaches the framework of claim 1. Specifically, the Examiner has stated that Livingston teaches a mapping mechanism in communication with a data handler mechanism that obtains a command object. It is respectfully submitted that both Livingston and Gibbs teach of a graphical user interface, which is not equivalent to a data handler mechanism. A data handler mechanism, as required by claim 1, is passed data from a data retriever mechanism which obtains the data on which a command object is arranged to operate. A graphical user interface, on the other hand, is arranged to obtain user input. In other words, according to the Examiner's argument that a graphical user interface is equivalent to a data handler mechanism, a user would have to input the data on which a command object is to operate. As such, a data handler mechanism that communicates with a data retriever mechanism is not identical to a graphical user interface that interfaces with a user. Hence, the Applicants submit that a data handler mechanism is not equivalent to a graphical user interface.

The Applicants note that the Examiner has asserted that a folder is equivalent to a data retrieval mechanism, and that a graphical user interface is passed data from a folder. It is respectfully submitted that a folder is not arranged to obtain data but is, instead, used to store data (Gibbs, page 45). A folder does not perform any "active" functions like obtaining data and passing data. That is, a folder in not capable of obtaining data or passing data. Instead, a folder is used to store data that may be used by mechanism, *e.g.*, a folder stores data such that the data may be passed to a data retriever mechanism by passing the folder. The Applicants submit that neither Gibbs nor Livingston teaches or suggests a data retrieval mechanism which is in communication with a data handler such that it obtains data and passes the data to the data handler. Therefore, claim 1 is believed to be allowable for at least this reason.

In addition, the graphical user interfaces taught by both Livingston and Gibbs appear to be a part of a program. For instance, the graphical user interface of Gibbs is part of an e-mail program (Gibbs, page 63). Such a graphical user interface does not interface with an application but is, instead, a part of the application. As stated above, a graphical user interface is not equivalent to a data handler mechanism. However, it is noted that while a data handler mechanism interfaces with an application, the graphical user interface does not interface with an application but is, instead, part of an application. As such, claim 1 is believed to be allowable over Livingston and Gibbs for at least this additional reason.

Further, claim 1 requires a mapping mechanism that obtains a command object based substantially on the data on which the command object is to operate. Livingston does not teach or reasonably suggest such a mapping mechanism. According to the Examiner, in teaching of a registry that is in communication with a graphical user interface, Livingston teaches of a mapping mechanism that is in communication with a data handler mechanism and is arranged to obtain a command object. It is respectfully submitted that the registry as taught by Livingston provides a list of actions that an application can undertake (Livingston, page 262). However, the registry of Livingston does not actually undertake any actions and, hence, does not obtain a command object based substantially on the data on which the command object is to operate. In addition, Livingston does not teach or reasonably suggest that the actions listed in a registry are undertaken based substantially on data on which the actions are to operate. Instead, Livingston discloses that actions in the registry require some user input (Livingston, page 263). In other words, according to Livingston, the choice of an action is based upon a user input, and is not based on the data on which the action is to operate. Therefore, since neither Livingston nor Gibbs, alone or in combination, teaches or suggests a mapping mechanism which obtains a command object based substantially on the data on which the command object is arranged to operate, claim 1 is believed to be allowable for at least this reason as well.

Claims 2-8 each depend either directly or indirectly from independent claim 1, and are, therefore, each believed to be allowable over the art of record for at least the reasons set forth with respect to claim 1. Each of these dependent claims recites additional limitations which, when considered in view of independent claim 1, further distinguish the claimed invention over the art of record.

In view of the above, Applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

If any fees are due in connection with the filing of this amendment, the Commissioner is authorized to charge such fees to Deposit Account 50-0388 (Order No. SUN1P123). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
BEYER & WEAVER, LLP

Peggy A. Su

Peggy A. Su
Registration No. 41,336

BEYER & WEAVER, LLP
P.O. Box 61059
Palo Alto, CA 94306
Tel: (650) 493–2100